SYSLOG-1

Always bound input to avoid internal buffer overflow.

Sean Barnum, Cigital, Inc. [vita[1]]

2007-04-23

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 3639 bytes

<table>
<tr><td>Attack Category</td><td>• Malicious Input</td></tr>
<tr><td>Vulnerability Category</td><td>• Buffer Overflow
• Unconditional</td></tr>
<tr><td>Software Context</td><td>• Logging</td></tr>
<tr><td>Location</td><td></td></tr>
<tr><td>Description</td><td>syslog() has internal buffer limitations, so size of input should be bounded.

syslog() is used to log system messages. It has internal buffer limitations that are implementation dependent.</td></tr>
<tr><td>APIs</td><td><table>
<tr><th>Function Name</th><th>Comments</th></tr>
<tr><td>syslog()</td><td></td></tr>
<tr><td>openlog()</td><td>Presence of openlog() is an indicator of nearby syslog() calls</td></tr>
</table></td></tr>
<tr><td>Method of Attack</td><td>On some platforms, an internal buffer can overflow in syslog(). If an attacker can control this they can mount a buffer overflow attack.</td></tr>
<tr><td>Exception Criteria</td><td></td></tr>
<tr><td>Solutions</td><td><table>
<tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr>
<tr><td>Whenever syslog() is used.</td><td>Determine whether your platform has syslog bugs. Limit the amount/size of information going into syslog.</td><td>Effective if accurate information is available on implementation.</td></tr>
</table></td></tr>
<tr><td>Signature Details</td><td>void syslog(int priority, char *format, ...)</td></tr>
</table>

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

Examples of Incorrect Code	`[...]`
	`openlog("my_program", 0, LOG_USER);`
	`void syslog( LOG_WARNING, "Ran into big problem with data: %s", veryLongText);`
	`closelog();`
	`[...]`

Examples of Corrected Code	`[...]`
	`openlog("my_program", 0, LOG_USER);`
	`/* Truncate text to ensure not too big. If syslog() formatting supports it, could alternatively use formatting options that limit output size. */`
	`char textBuffer[MAX_SAFE_TEXT_SIZE]; /* Safe size depends on syslog() implementation */`
	`strncpy(textBuffer,veryLongText,sizeof(textB`
	`textBuffer[sizeof(textBuffer)-1] = '\0';`
	`void syslog( LOG_WARNING, "Ran into big problem with data: %s", textBuffer);`
	`closelog();`
	`[...]`

Source Reference	• Viega, John & McGraw, Gary. *Building Secure Software: How to Avoid Security Problems the Right Way*. Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X, p. 148.
Recommended Resource	• Syslog man page[2]
Discriminant Set	**Operating System**: • Windows **Languages**: • C • C++

Cigital, Inc. Copyright

1. mailto:copyright@cigital.com

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.